

## **Sistema de Reservas de Habitaciones de un Hotel.**

### **Enunciado.**

Desarrollar un sistema de gestión reservas de habitaciones de un Hotel utilizando Java. El sistema debe permitir a los clientes realizar reservas de habitaciones estándar y suites, gestionar las reservas de manera eficiente y facilitar la administración de las habitaciones disponibles. La solución debe ser robusta y eficiente, utilizando las mejores prácticas de desarrollo orientado a objetos.

El diseño debe seguir los principios de la orientación a objetos, asegurando la cohesión y el bajo acoplamiento de las clases. Se tendrá en cuenta el correcto empleo de los conceptos de herencia, clases abstracta, polimorfismo, interface, colecciones avanzadas, casos de prueba y manejo adecuado de excepciones para garantizar la integridad y robustez del sistema.

El sistema debe permitir:

- Agregar habitaciones estándar y suites al inventario del hotel. Cada habitación se identificará de manera única por su número de habitación.
- Realizar reservas para habitaciones disponibles. La disponibilidad debe ser verificada en tiempo real para evitar conflictos.
- La cancelación de reservas existentes, liberando las habitaciones para nuevas reservas.
- Consultar las reservas existentes. La consulta debe ser rápida y eficiente.
- Manejo de Excepciones. Debe manejar adecuadamente situaciones como intentar reservar una habitación que ya está ocupada o que no existe, mediante el uso de excepciones específicas.

Se pide:

- Implementar las clases necesarias que den solución al problema.
- Generen al menos los siguientes casos de prueba que verifiquen la funcionalidad deseada.

1. Agregar una Habitación Estándar al Sistema:

*Entrada:* Información de la habitación estándar (número de habitación, capacidad, precio por noche, tiene balcón, vista, incluye desayuno, TV por cable).

*Resultado Esperado:* La habitación se agrega al sistema y se puede recuperar correctamente.

2. Realizar una Reserva en una Habitación Estándar Disponible:

*Entrada:* Información del cliente (nombre, documento de identidad), número de habitación, fecha de inicio y fin de la reserva.

*Resultado Esperado:* La reserva se realiza con éxito y se puede recuperar la información de la reserva.

3. Cancelar una Reserva:

*Entrada:* Número de habitación.

*Resultado Esperado:* La reserva se cancela y la habitación queda disponible nuevamente.

4. Intentar Reservar una Habitación Inexistente:

*Entrada:* Información del cliente (nombre, documento de identidad), número de habitación inexistente, fecha de inicio y fin de la reserva.

*Resultado Esperado:* Se lanza una excepción "HabitacionInexistenteException" con el mensaje "La habitación no existe."

5. Intentar Reservar una Habitación ya Ocupada:

*Entrada:* Información del cliente (nombre, documento de identidad), número de habitación, fecha de inicio y fin de la reserva (mismo periodo que una reserva existente).

*Resultado Esperado:* Se lanza una excepción "HabitacionOcupadaException" con un mensaje indicando que la habitación está ocupada.

6. Agregar una Suite al Sistema y Realizar una Reserva:

*Entrada:* Información de la suite (número de habitación, capacidad, precio por noche, tamaño de la suite, tiene jacuzzi, servicio de mayordomo, zona de estar, número de habitaciones).

*Resultado Esperado:* La suite se agrega al sistema y se puede realizar una reserva con éxito.

7. Verificar Atributos Específicos de una Habitación Estándar y una Suite:

*Entrada:* Recuperar información de una habitación estándar y una suite.

*Resultado Esperado:* Los atributos específicos (por ejemplo, tiene balcón, vista, incluye desayuno, TV por cable para la estándar; tamaño de la suite, tiene jacuzzi, servicio de mayordomo, zona de estar, número de habitaciones para la suite) son correctos.